REMARKS

In response to the Office Action dated September 22, 2003, the Applicant is filing this Amendment and Response. At the time of the Office Action, claims 1-52 were pending. In this Amendment and Response, claim 20 is amended. No new claims are being added and no claims are being canceled. Accordingly, claims 1-52 remain currently pending.

In the Office Action, the Examiner objected to the drawings because of a missing reference numeral in FIG. 2. The Applicant is providing a replacement drawing for FIG. 2 (attached as Appendix A hereto), which includes the missing reference numeral. Accordingly, the Applicant respectfully requests withdrawal of the objection to the drawings.

In addition, claim 20 was objected to because of an apparent mislabeling of one of the steps. Claim 20 has been amended to correct this informality. Accordingly, the Applicant respectfully requests withdrawal of the objection to claim 20.

Claims 1-4, 7-10, 13-17, 20-25, 28-33, 36-40, 43-48, 51 and 52 were rejected under 35 U.S.C. § 102(a) as being anticipated by U.S. Patent No. 5,974,455 to Monier ("the Monier reference"). Additionally, claims 5, 6, 11, 12, 18, 19, 26, 27, 31, 32, 34, 35, 49 and 50 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Monier in view of U.S. Patent No. 5,953,729 to Cabrera ("the Cabrera reference"). Each of the rejections is addressed in detail below.

Rejections under 35 U.S.C. § 102

As set forth above, the Examiner rejected claims 1-4, 7-10, 13-17, 20-25, 28-33, 36-40, 43-48, 51 and 52 under 35 U.S.C. § 102(a) as being anticipated by the Monier reference. Specifically, the Examiner stated:

> 4. Claims 1-4, 7-10, 13-17, 20-25, 28-33, 36-40, 43-48, 51 and 52 are rejected under 35 U.S.C. 102(a) as being anticipated by Monier (U.S. Patent No. 5,974,455).
>
> Monier teaches the invention as claimed including a method, a computer program and a web crawler system for efficient representation of data set addresses in a web crawler (see Monier. Abstract).
>
> 5. In reference to claims 1, 13, 22, 23, 31, 38 and 46, Monier teaches downloading data sets from among a plurality of host computers comprising the following steps:
>
> Storing representations of data set addresses in a set of data structures, including a first buffer, a second buffer and a first disk file, wherein representations of data set addresses stored in the first disk file are ordered (column 3, lines 1-35, Monier discloses storing URL representations in a set of data structures, including a hash table (stored in random access memory (RAM)), an append buffer (stored in RAM) and a sequential disk file, wherein the representations are stored sequentially in the disk file).
>
> Selecting as a current buffer one of the first and second buffers (column 6. lines 35-45, Monier discloses selecting and managing a current buffer among the hash table and append buffer).
>
> Downloading at least one data set that includes addresses of one or more referred data sets (column 5, lines 20-30, Monier discloses fetching web pages that include URL's of one or more referred web pages).
>
> Identifying the addresses of the one or more referred data sets (column 5, lines 20-30, Monier discloses analyzing and identifying the addresses of the one or more referred web pages).
>
> For each identified address:

Generating a representation of the identified address (column 5 line 55 -- column 6 line 22, Monier discloses generating a fingerprint representation of the specified URL), and Determining whether the representation is stored in the buffer, and when this determination is negative, storing the representation in the buffer (column 5 line 43 -column 6 line 22, Monier discloses determining whether the representation is stored in the hash table, and when this determination is negative, storing the representation in the hash table).

When the buffer reaches a predefined full condition:

Ordering the contents of the buffer according to the representations (column 6, lines 1-33, Monier discloses ordering the contents of the hash table according to the fingerprint representations), and

Performing an ordered merge of the contents of the buffer into the contents of the first disk file (column 6 line 22 -column 7 line 12, Monier discloses performing a merge of the contents of the hash table into the contents of the disk file), and

Selecting the other buffer as the current buffer, wherein the previously current buffer is identified as a non-current buffer (column 6, lines 22-67, Monier discloses selecting the append buffer as the current buffer, wherein the hash table is identified as a non-current buffer).

6. In reference to claims 2, 14, 24, 32, 39 and 47, Monier teaches the method, the computer program and the web crawler system of claims 1, 13, 22, 23, 31, 38 and 46 above, wherein after determining that the representation is not stored in the buffer, the identified address is stored in the buffer (column 5 line 43 -column 6 line 22, Monier discloses that after detem1ining that the representation is not stored in the hash table, the identified address is stored in the hash table).

7. In reference to claims 3, 15, 25, 33, 40, 48, Monier teaches the method, the computer program and the web crawler system of claims 1, 13, 23, 31, 38 and 46 above, wherein:

After determining that the representation is not stored in the buffer, the identified address is stored in a second disk file (column 9, lines 25-40, Monier discloses after determining that the representation is not stored in the hash table, the identified address is stored in a second disk file, and

Addtiona1 1y storing with each representation in the buffer- a pointer to the corresponding address stored in the second disk file {column

3, lines 1-20, column 5 & column 6, lines 20-53, Monier discloses additionally storing with each representation in the RAM a pointer to the corresponding address stored in the second disk file), and

While ordering the contents of the buffer, keeping with each representation in the buffer its pointer to the corresponding address in the second disk file (column 5 & column 6, lines 20-53, Monier discloses while ordering the contents of the hash table (in RAM), keeping with each representation in the hash table its pointer to the corresponding address in the disk file).

8. In reference to claims 4 and 16, Monier teaches the method of claims 3 and 15 above, wherein when the buffer reaches a predefined full condition:

Each representation in the buffer stores an associated flag, setting the flag to a first value when the representation is equal to a representation previously stored in the first disk file, and setting the flag to a second value, when the representation is not equal to any representation previously stored in the first disk file (column 5 lines 25-35, & column 8 lines 45-65, Monier discloses each representation in the hash table stores an associated " fetched flag", setting the flag to a first value when the representation is equal to a representation previously stored in the disk file, and setting the flag to a second value, when the representation is not equal to any representation previously stored in the disk file), and

Each representation whose flag is set to the second value, scheduling the corresponding data set for downloading (column 9. lines 25-50, Monier discloses each representation whose flag is set to the second value and marked as "not fetched", scheduling the corresponding data set for fetching).

9. In reference to claims 7, 20, 28, 36, 43 and 51, Monier teaches the method, the computer program and the web crawler system of claims 1, 13, 23, 31, 38 and 46 above, wherein the representation of the identified address comprises a checksum of at least a portion of the identified address (column 5 line 55 -column 6 line 22, Monier discloses the representation of the identified URL comprising a fingerprint of at least a portion of the identified URL).

10. In reference to claims 8, 21, 29 and 44, Monier teaches the method, the computer program and the web crawler system of claims 1, 13, 23 and 38 above, wherein:

Determining whether the representation is stored in a cache before determining whether the representation is stored in the

buffer (columns 6&7, Monier discloses determining whether the representation is stored in append buffer before determining whether the representation is stored in an input buffer (in RAM)), and

Determining whether the representation is stored in a cache, and if positive, skipping the determination of whether the representation is stored in the buffer (columns 6&7, Monier discloses determining whether the representation is stored in an append buffer, and if positive, skipping the determination of whether the representation is stored in the input buffer), and

When the representation is not stored in the cache, the cache has not reached a predefined full condition, and other predefined criteria are met, adding the representation to the cache (columns 6&7, Monier discloses when the representation is not stored in the append buffer, the host name table has not reached a predefined full condition, and other predefined criteria are met, adding the representation to the input buffer), and

When the representation is not stored in the cache, the cache has reached said predefined full condition, and said other predefined criteria are met, evicting a stored representation from the cache in accordance with an eviction policy and adding the representation to the cache (columns 6&7, Monier discloses when the representation is not stored in the append buffer, the append buffer has reached said predefined full condition, and said other predefined criteria are met, evicting a stored representation from the append buffer in accordance with an eviction policy and adding the representation to the append buffer).

11. In reference to claims 9, 10, 17, 30, 37, 45 and 52, Monier teaches the method, the computer program and the web crawler system of claims 1, 23, 31, 38 and 46 above, wherein when a representation in the first buffer is not found in the first disk file during merging, scheduling the corresponding data set for downloading (columns 6-8, Monier discloses that when a representation in the hash table is not found in the disk file during merging, scheduling the corresponding web page for fetching).

Office Action, pages 2-7.

The Applicant respectfully traverses this rejection. Anticipation under section 102 can be found only if a single reference shows exactly what is claimed. *Titanium Metals Corp. v. Banner*, 778 F.2d 775, 227 U.S.P.Q. 773 (Fed. Cir. 1985). For a prior art reference to anticipate

under section 102, every element of the claimed invention must be identically shown in a single reference. *In re Bond*, 910 F.2d 831, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990). To maintain a proper rejection under section 102, a single reference must teach each and every element or step of the rejected claim. *Atlas Powder v. E.I. du Pont*, 750 F.2d 1569 (Fed. Cir. 1984). Accordingly, the Applicant needs only to point to a single element not found in the cited reference to demonstrate that the cited reference fails to anticipate the claimed subject matter. The prior art reference also must show the *identical* invention "*in as complete detail as contained in the ... claim*" to support a *prima facie* case of anticipation. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q. 2d 1913, 1920 (Fed. Cir. 1989).

In the present case, the Monier reference cannot anticipate Applicant's claims under Section 102 because every element of the Applicant's claimed invention is not identically shown in Monier. The Applicant's invention relates to a method and apparatus for updating a list of URLs with newly discovered URLs identified during a web crawl operation. As new URLs are identified, they are added to a buffer under certain conditions. Subsequently, the contents of the buffer may be processed and incorporated into a disk file.

Independent method claim 1 requires that, when the buffer reaches a predefined full condition: "(el) *ordering the contents of the buffer according to the representations*, and (e2) *performing an ordered merge* of the contents of the buffer into the contents of the first disk file." (Emphasis added). Independent method claim 13 requires that, when one of two buffers is full, (f2) *ordering the representations stored in the non-current buffer*, and (f3) *performing an ordered merge* of the contents of the non-current buffer into the contents of the first disk file. (Emphasis added). Independent claim 23 is directed to a computer program that includes

instructions for "determining whether the buffer has reached a predefined full condition, and when this determination is positive, *ordering the contents of the buffer* and then *performing an ordered merge* of contents of the buffer into the contents of the first disk file." (Emphasis added). Independent claim 31 is directed to a computer program that "*order[s] the contents of the non-current buffer* and then *perform[s] an ordered merge* of the contents of the non-current buffer into the contents of the first disk file." (Emphasis added).

Independent claim 38 is directed to a web crawler containing an address filtering module that "*order[s] the contents of the buffer* and then *perform[s] an ordered merge* of the contents of the buffer into the contents of the first disk file." (Emphasis added). Independent claim 46 is directed to a web crawler including an address filtering module that "*order[s] the contents of the non-current buffer* and then *perform[s] an ordered merge* of the contents of the non-current buffer into the contents of the first disk file." (Emphasis added).

The recited limitations are not disclosed in Monier. Monier only appears to disclose the appending of data collected in an append buffer to the end of a disk file. The operation of the append buffer is explained in Monier, as follows:

> All accesses to the Web information disk file 150 are made sequentially via the input buffer 134 such that a large number of entries from the sequential disk file are moved into the input buffer as single I/O operation. The sequential disk file 150 is then accessed from the input buffer. Similarly, all new entries to be added to the sequential file are stored in the append buffer 136, and *the contents of the append buffer are added to the end of the sequential whenever the append buffer is filled.* In this way random access to the Web information disk file is eliminated, and latency caused by disk access limitations is minimized.

> Each time all the entries in the input buffer 134 have been scanned by the Web scooter, all updates to the entries in the input buffer are stored back into the Web information disk file 150 and *all entries in the append buffer 136 are appended to the end of the disk file 150.* In addition, the append buffer 136 is cleared and the next set of entries in the disk file, starting immediately after the last set of entries to be copied into the input buffer 134 (as indicated by pointer 178), are copied into the input buffer 134. When the last of the entries in the disk file have been scanned by the Web scooter procedure, scanning resumes at the beginning of the disk file 150.
>
> Whenever the append buffer 136 is filled with new entries, *its contents are appended to the end of the disk file 150* and then the append buffer is cleared to receive new entries.
>
> Monier, col. 6, line 53 – col. 7, line 11 (Emphasis added).

Thus, Monier does not contemplate additional processing of information that is obtained during a web crawl operation and stored in the append buffer. Monier, therefore, cannot anticipate independent claims 1, 13, 23, 31, 38 and 46, and the claims dependent thereon.

The Examiner improperly equates the ordering operation of the newly discovered URL information to the processing of data stored in a hash table in Monier. In the Monier reference, the hash table is loaded sequentially from the disk file prior to starting a web crawl operation. That data may be in the form of a fingerprint and may be stored in order. The comparison of the processing of the hash data to the operation of the Applicant's claimed invention is not correct because the hash data is not subsequently processed and stored on the disk via an ordered merge, as recited in claims 1, 13, 23, 31, 38 and 46, and the claims dependent thereon. In Monier, the hash data is processed in sequential order because it is retrieved sequentially from hard drive storage. Monier does not discuss processing of information *added to the append buffer* during a web crawl operation, other than to say that the information in the append buffer is appended

directly to the end of the disk file. For at least these reasons, the Examiner's rejection of claims 1, 13, 23, 31, 38 and 46, and the claims dependent thereon is improper and should be withdrawn.

Independent method claim 22 recites the acts of "determining whether the representation is stored in the buffer, and whether the disk file is empty", "when the representation is not stored in the buffer and the disk, file is empty scheduling the corresponding data set for downloading", and "when the representation is not stored in the buffer and the disk file is .not empty, storing the representation in the buffer and delaying scheduling of the corresponding data set for downloading until it is determined that the representation has not been previously stored in the disk file." Monier contains no teaching, suggestion or illustration that relates to deciding what to do with data based on whether the buffer or disk file is empty. In the text of the rejection of claim 22, the Examiner erroneously treats claim 22 as though it contained limitations similar to the limitations previously discussed. Monier does not contain those limitations at all. For at least these reasons, Monier cannot anticipate independent method claim 22. The rejection of claim 22 under Section 102 based on Monier should, therefore, be withdrawn.

Rejections under 35 U.S.C. § 103

The Examiner rejected claims 5, 6, 11, 12, 18, 19, 26, 27, 31, 32, 34, 35, 49 and 50 under 35 U.S.C. § 103(a) as being obvious based on Monier in view of Cabrera. Specifically, the Examiner stated:

13. Claims 5, 6, 11, 12, 18, 19, 26, 27, 34, 35, 31, 32, 49 and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Monier (U.S. Patent No. 5,974,455) in view of Cabrera et al. (U.S. Patent No. 5,953,729).

14. In reference to claims 5, 11, 18, 26, 34, 41 and 49, Monier teaches the method, the computer program and the web crawler system of claims 1, 13, 23, 31, 38 and 46 above.

Monier does not teach storing representations of data set addresses in a sparse disk file which is divided into portions (or sub-files), each portion having a starting address and contents comprising an ordered list of representations of data addresses. However, Cabrera teaches sparse file technology divided into clusters each having a cluster number (column 9, lines 40-66).

It would have been obvious to one having ordinary skill in the art to modify Monier by storing URL representations in a sparse file as per the teachings of Cabrera so as to minimize the overhead in managing and ordering the contents on the disk file.

15. Monier does not teach merging the contents of the buffer with the ordered contents of the sparse disk file to include determining a starting address for a corresponding portion of the sparse disk file. However, Cabrera teaches sparse file technology which can indicate starting cluster numbers for portions of the sparse file (columns 9& 10).

It would have been obvious to one having ordinary skill in the art to modify Monier by when merging the contents of the hash table with the ordered contents of the sparse file, to include determining a starting cluster number for a corresponding portion of the sparse disk file as per the teachings of Cabrera so as to minimize the overhead for merging and ordering of the contents on the disk file.

16. Monier does not teach merging the contents of the buffer with the ordered contents of the sparse disk file to include performing an ordered merge of a subset of the buffer, starting at the representation for which the starting address was obtained, into the contents of the corresponding portion. However, Cabrera teaches sparse file technology which can indicate starting cluster numbers for portions of the sparse file (columns 9&10).

It would have been obvious to one having ordinary skill in the art to modify Monier by when merging the contents of the hash table with the ordered contents of the sparse disk file to include

performing an ordered merge of a subset of the hash table, starting at the representation for which the starting address was obtained, into the contents of the corresponding portion as per the teachings of Cabrera so as to minimize the overhead in merging and ordering the contents on the disk file.

17. In reference to claims 6, 12, 19, 27, 35, 42 and 50, Monier teaches the method, the computer program and the web crawler system of claims 1, 13, 23, 31, 38 and 46 above.

18. Monier does not teach storing representations of data set addresses in a sparse disk file having empty entries interspersed among entries storing said representations. However, Cabrera teaches sparse file technology which comprises a mixture of zero data and non-zero data (column 7, lines 20-50).

It would have been obvious to one having ordinary skill in the art to modify Monier by storing representations of data set addresses in a sparse disk file having zero data interspersed among data of said representations as per the teachings of Cabrera so as to minimize the overhead in sequentially ordering the data contents on the disk file.

19. Monier teaches sequentially scanning the disk file via an input buffer, starting at the representation for which a starting address was obtained, until a representation matching the respective representation is found (column 6 lines 35-67 & column 9 lines 25-50). Monier does not teach scanning the disk until one of the empty entries is found, and when an empty entry is found storing the respective representation in the empty entry. However, Cabrera teaches sparse file technology which comprises a mixture of zero data and non-zero data (column 7. lines 20-50).

It would have been obvious to one having ordinary skill in the art to modify Monier by scanning the disk file until one of the zero data entries is found as per the teachings of Cabrera, and when zero data entry is found storing the respective representation in the zero data entry, so as to minimize the overhead of ordering the data contents on the disk file while merging the contents of the hash table with the contents of the disk file.

Office Action, pages 7-10.

The Applicant respectfully traverses this rejection. The burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (PTO Bd. App. 1979). Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teaching or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Accordingly, to establish a *prima facie* case, the Examiner must not only show that the combination includes *all* of the claimed elements, but also a convincing line of reason as to why one of ordinary skill in the art would have found the claimed invention to have been obvious in light of the teachings of the references. *Ex parte Clapp*, 227 U.S.P.Q. 972 (B.P.A.I. 1985). When prior art references require a selected combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gained from the invention itself, i.e., something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination. *Uniroyal Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 5 U.S.P.Q.2d 1434 (Fed. Cir. 1988).

Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teachings or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital,* 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed Cir. 1984). One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention. *In re Fine,* 837 F.2d 1071, 5 U.S.P.Q. 2d 1596 (Fed. Cir. 1988).

In the present case, the combination of Monier and Cabrera cannot render the Applicant's claims obvious under Section 103 because Monier and Cabrera do not disclose all of the

elements recited in the Applicant's claims. At least the elements set forth above with respect to the rejection under Section 102 of independent claims 1, 13, 23, 31, 38 and 46 are entirely missing from Monier. Those elements are not supplied by Cabrera. Indeed, they are not alleged to be found in Cabrera by the Examiner. For at least these reasons, the Applicant respectfully requests withdrawal of the rejection of claims 5, 6, 11, 12, 18, 19, 26, 27, 31, 32, 34, 35, 49 and 50 under Section 103 based on the combination of Monier and Cabrera.

Conclusion

In view of the Amendments and Remarks set forth above, the Applicant respectfully requests withdrawal of all of the Examiner's objections and rejections. Furthermore, the Applicant asserts that an indication of the allowability of claims 1-52 is appropriate. If the Examiner believes that a telephonic interview will help speed this application toward issuance, the Examiner is invited to contact the undersigned at the telephone number listed below.

Respectfully submitted,

Date: December 22, 2003

Barry D. Blount
Reg. No. 35,069
FLETCHER YODER
P.O. Box 692289
Houston, TX 77269-2289
(281) 970-4545

CORRESPONDENCE ADDRESS:

Intellectual Property Administration
Legal Department, M/S 35
HEWLETT-PACKARD COMPANY
P.O. Box 272400
Fort Collins, CO 80527-2400